NUI Galway
OÉ Gaillimh

Autumn Examinations 2014/ 2015

Exam Code(s)	4BCT, 3BA, 1SD, 1MDM
Exam(s)	4th Year B.Sc. (CS&IT) 3rd Year B.A. (Information Technology) Higher Diploma in Applied Science (Software Design & Development) Masters in Digital Media
Module Code(s)	CT404, CT336
Module(s)	Graphics and Image Processing
Paper No.	
Repeat Paper	
External Examiner(s)	Prof. L. Maguire Dr. J. Power
Internal Examiner(s)	Prof. G. Lyons Dr. M. Madden * Dr. S. Redfern
Instructions:	Answer any three questions. All questions carry equal marks.
Duration	2 hours
No. of Pages	5
Discipline(s)	Information Technology
Course Co-ordinator(s)	

Requirements:

MCQ	Release to Library:	Yes	X	No
Handout				
Statistical/ Log Tables				
Cambridge Tables				
Graph Paper				
Log Graph Paper				
Other Materials				
Graphic material in colour	Yes	X	No	

PTO

Q.1. (Graphics)

(i) Many of the techniques used in real-time 3D graphics programming attempt to maximise the realism of the rendered scene while using a minimal number of polygons. With specific reference to this so-called 'polygon budget', and using diagrams where appropriate, discuss each of the following five techniques:

- Back-face culling [2]
- Texture mapping [2]
- Bump mapping [2]
- Skyboxes [2]
- Levels of Detail (LODs) [2]

(ii) Consider the Canvas/Javascript code shown below, which draws a purple rectangle, size 50x50 pixels onto a Canvas of size 600x600 pixels.

Explain the use of the following three lines of code: [6]

```
context.translate(x,y);
context.restore();
window.setTimeout("draw();",1000);
```

(iii) Modify the code so that the purple rectangle is redrawn at a random position on the canvas, once per second. The entire width and height of the canvas should be used. Hint: the javascript function `Math.random()` can be used to generate a random float between 0 and 1 [4]

```
<html>
 <head>
  <script>
var x=200, y=200;
function draw() {
  var canvas = document.getElementById("canvas");
  var context = canvas.getContext('2d');

  context.save();

  // over-write previous content, with a grey rectangle
  context.fillStyle="#DDDDDD";
  context.fillRect(0,0,600,600);

  context.translate(x,y);

  // draw a purple rectangle, size 50,50
  context.fillStyle="#CC00FF";
  context.fillRect(0,0,50,50);

  context.restore();

  window.setTimeout("draw();",1000);
}
  </script>
 </head>
 <body onload="draw();">
   <canvas id="canvas" width="600" height="600"></canvas>
 </body>
</html>
```

Q.2. (Graphics)

Figure 1 **Figure 2**

(i) Describe the use of extrusion in X3D, referring to each of the seven fields used by the Extrusion node. Note that extrusion and other useful nodes from the X3D language are summarised on the final page of this exam paper. [5]

(ii) Using extrusion, write X3D code to make the 3D model of the letter A, illustrated above (figure 1). Include a diagram illustrating your cross section points. [8]

(iii) Explain how you would modify your X3D code in order to bevel the edges of the letter A model, as illustrated in figure 2. [3]

(iv) In realtime 3D graphics, what are billboards, and how do they assist in the efficient rendering of an animated scene? [4]

Q.3. (Graphics)

(i) Discuss in general terms how graphics hardware has changed (improved) over the past number of years. In your answer, refer to these terms: Graphics Processing Unit (GPU), on-board RAM, texture memory, data bus, DirectX/OpenGL [5]

(ii) Why are nested co-ordinate systems useful for 3D graphics/animation programming? In your answer, explain and provide code samples illustrating the use of nested co-ordinate systems in both Javascript/Canvas and X3D. [5]

(iii) In 3D computer graphics, what is a surface normal? Illustrate your answer with a diagram. [4]

(iv) Discuss, with the use of a diagram in each case, the following 3D graphics techniques. In each case, identify the importance of surface normals:

a) Flat (Lambert) Shading
b) Normal Interpolating (Phong) Shading
c) Radiosity [6]

Q.4. (Image Processing)

(i) The first steps in many image processing algorithms often include blurring and edge enhancement. With reference to the noisy image below (Figure 3), explain how blurring and edge enhancement could assist with the extraction of the circles and oval from the image. [5]

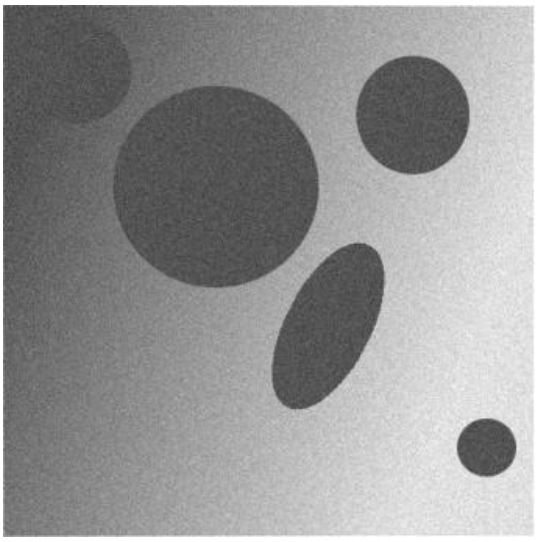

Figure 3

(ii) Describe, with use of a diagram, the convolution algorithm which is commonly used for both blurring and edge enhancement. Present a convolution kernel suitable for edge detection, and explain technically how this kernel produces the desired result, assuming a greyscale image with pixel values in the range 0 to 255. [7]

(iii) The Hough Transform (HT) is an image processing technique that detects simple geometric shapes in edge-enhanced images. Outline the HT algorithm for detecting circles, and sketch the results you might expect when this algorithm is applied to an edge-enhanced version of Figure 3. How might you algorithmically detect the oval shape, given data output from the HT for circles? [8]

Q.5. (Image Processing)

(i) Describe the morphological image processing techniques of erosion and dilation. Compare the four operations (i) opening, (ii) closing, (iii) thinning and (iv) thickening. For each of these 4 techniques, describe one situation where it would provide useful results. [10]

(ii) Consider the image of a bubble, shown in Figure 4

The image contains substantial amounts of noise, and there exists a large section of bright 'shine' pixels across the centre of the bubble. Propose and justify a series of image processing steps that would be suitable to accurately measure the number of pixels inside the bubble. [10]

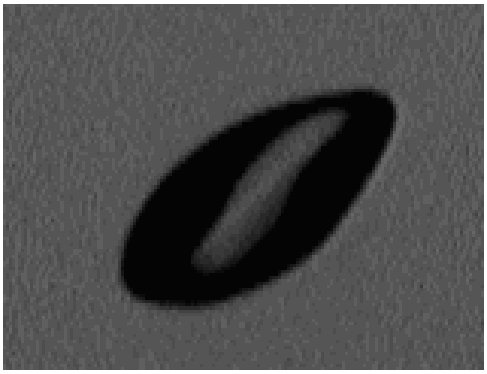

Figure 4

Some useful X3D nodes:

Node	Important Fields and Nested Nodes
Shape	Nested Nodes: Appearance, Geometry Nodes (Box, Sphere, Cone, Cylinder, Text, Extrusion, etc.)
Appearance	Nested Nodes: Material, ImageTexture
Material	Fields: diffuseColor, specularColor, emissiveColor, ambientIntensity, transparency, shininess
ImageTexture	Fields: url
Transform	Fields: translation, rotation, scale, center. Nested Nodes: Other Transforms, Shapes, Sensors
TimeSensor	Fields: enabled, startTime, stopTime, cycleInterval, loop
PositionInterpolator	Fields: key, keyValue
OrientationInterpolator	Fields: key, keyValue
Extrusion	Fields: crossSection, spine, scale, orientation, beginCap, endCap, creaseAngle
Box	Fields: size
Sphere	Fields: radius
Cylinder	Fields: radius, height, side, top, bottom
Cone	Fields: height, bottomRadius, side, bottom
PointLight	Fields: on, location, radius, intensity, ambientIntensity, color, attenuation
ROUTE	Fields: fromNode, fromField, toNode, toField

Some useful methods/properties of the Canvas 2D Context object:

Method/Property	Arguments/Values	Notes
fillRect	(Left, Top, Width, Height)	Draw a filled rectangle
beginPath	None	Start a stroked path
moveTo	(X, Y)	Move the graphics cursor
lineTo	(X, Y)	Draw a line from graphics cursor
stroke	None	End a stroked path
fillStyle	="rgb(R,G,B)"	Set fill colour
strokeStyle	="rgb(R,G,B)"	Set line colour
save	None	Save the current coordinate system
restore	None	Restore the last saved coord system
translate	(X,Y)	Translate the coordinate system
rotate	(angle)	Rotate the coordinate system clockwise, with angle in radians
scale	(X,Y)	Scale the coordinate system independently on the X and Y axes